Amendments to the claims (this listing replaces all prior versions):

1. (Currently Amended) A method comprising:

for each unit of data transferred between a first processor and a second processor,

filling a cache set of a cache memory with data associated with addresses in a main memory that correspond to the cache set, either

after the first processor writes a unit of data to addresses that correspond to the cache set or

before the first processor reads a unit of data written by the second processor to addresses that correspond to the cache set,

with the data used to fill the cache set being associated with addresses that are different from the addresses associated with the unit of data.

2. (Original) The method of claim 1 wherein filling the cache set with data comprises performing read operations to read data from addresses that are different from the addresses of the unit of data written by the second processor.

3. (Original) The method of claim 2 wherein performing read operations comprises using the first processor to perform the read operations.

4. (Original) The method of claim 2 wherein each cache set comprises a predetermined number of cache lines, wherein the number of read operations performed to read data to fill the cache set is equal to the predetermined number of cache lines.

5. (Original) The method of claim 1 wherein each cache set comprises a predetermined number of cache lines, and filling the cache set comprises writing data to at least a portion of each cache line in the cache set.

6. (Original) The method of claim 1 wherein each unit of data has a size that corresponds to a cache line of the cache memory.

7. (Original) The method of claim 1 wherein the first processor has access to the cache memory.

8. (Original) The method of claim 1 wherein the second processor does not have access to the cache memory.

9. (Original) The method of claim 1 wherein each cache set of the cache memory is associated with particular addresses in the main memory.

10. (Original) The method of claim 1, further comprising using the second processor to write a unit of data to addresses in the main memory that correspond to the cache set, and notifying the first processor that a unit of data has been written to the main memory.

11. (Original) The method of claim 1, further comprising using a memory management unit to evict cached data associated with the same addresses as the unit of data after performing the read operations reading data associated with addresses that correspond to the same cache set as the addresses of the unit of data.

12. (Original) The method of claim 1, further comprising using the first processor to write a unit of data to addresses that correspond to the cache set, the unit of data being stored in the cache set before the cache set is filled with data associated with different addresses that also correspond to the cache set.

13. (Original) The method of claim 1, further comprising allocating a first portion of the main memory so that it does not store data transferred between the first and second processors.

14. (Original) The method of claim 13, further comprising allocating a second portion of the main memory to store data transferred from the first processor to the second processor.

15. (Currently Amended)The method of claim 14, further comprising reading data from the first portion after ~~to~~ using the first processor to write data to addresses corresponding to the second portion.

16. (Currently Amended) The method of claim ~~14~~ 13, further comprising allocating a ~~third~~ second portion of the main memory to store data transferred from the second processor to the first processor.

17. (Original) The method of claim 16, further comprising reading data from the first portion prior to using the first processor to read data from the ~~third~~ second portion.

18. (Currently Amended) A machine-accessible medium, which when accessed results in a machine performing operations comprising:

for each unit of data transferred between a first processor and a second processor, filling a cache set of a cache memory with data associated with addresses in a main memory that correspond to the cache set either

after the first processor writes a unit of data to addresses that correspond to the cache set or

before the first processor reads a unit of data written by the second processor to addresses that correspond to the cache set,

with the data used to fill the cache set being associated with addresses that are different from the addresses associated with the unit of data.

19. (Original) The machine-accessible medium of claim 18 wherein filling the cache set with data comprises performing read operations to read data from addresses that are different from the addresses of the unit of data written by the second processor.

20. (Original) The machine-accessible medium of claim 18, which when accessed results in the machine allocating a first portion of the main memory so that it does not store data transferred between the first and second processors.

21. (Original) The machine-accessible medium of claim 20, which when accessed results in the machine performing operations further comprising reading data from the first portion after using the first processor to write data intended for the second processor.

22. (Original) The machine-accessible medium of claim 20, which when accessed results in the machine performing operations further comprising reading data from the first portion prior to using the first processor to read data written by the second processor.

23. (Original) An apparatus comprising:

a first processor that accesses a main memory and a cache memory, the cache memory being divided into cache sets, each cache set corresponding to predefined addresses in the main memory; and

a second processor that accesses the main memory directly without accessing the cache memory, with the first processor generating dummy read instructions either

after generating one or more write instructions to write data intended for the second processor or

before generating one or more read instructions to read data that was written by the second processor, the dummy read instructions causing cached data to be evicted from a cache set.

24. (Original) The apparatus of claim 23 wherein the dummy read instructions comprise instructions to read data from addresses in the main memory that do not store data transferred between the first and second processors.

25. (Original) The apparatus of claim 23 wherein each cache set has n cache lines, n being an integer, and the first processor generates at least n dummy read instructions either after generating one or more write instructions to write data intended for the second processor or before generating one or more read instructions to read data that was written by the second processor.

26. (Original) The apparatus of claim 23 wherein the main memory comprises a first portion that does not store data transferred between the first and second processors.

27. (Original) The apparatus of claim 26 wherein the first portion of the main memory has a size that is at least the number of cache lines per cache set multiplied by the number of cache sets multiplied by the number of bytes per cache line.

28. (Original) A computer system comprising:

a main memory;

a cache memory that is divided into cache sets, each cache set having cache lines, each cache set corresponding to a predefined range of addresses in the main memory;

a first processor that accesses both the main memory and the cache memory;

a second processor that accesses the main memory directly without accessing the cache memory, with the first processor generating dummy read instructions either

after generating one or more write instructions to write data intended for the second processor or

before generating one or more read instructions to read data that was written by the second processor;

a memory management unit to control access of the cache memory by the first processor; and

a chipset to control access of the main memory by the first and second processors.

29. (Original) The computer system of claim 28, wherein the main memory comprises a first portion that does not store data transferred between the first and second processors.

30. (Original) The computer system of claim 28 wherein each cache set has n cache lines, n being an integer, and the first processor generates at least n dummy read instructions either

after generating one or more write instructions to write data intended for the second processor or

before generating one or more read instructions to read data that was written by the second processor.

31. (Original) A system comprising

a router to route data in a network;

a first processor that accesses a main memory and a cache memory, the cache memory being divided into cache sets, each cache set corresponding to predefined addresses in the main memory; and

a second processor to process data routed by the router, the second processor accessing the main memory directly without accessing the cache memory, with the first processor generating dummy read instructions either

after generating one or more write instructions to write data intended for the second processor or

before generating one or more read instructions to read data that was written by the second processor to the main memory, the dummy read instructions causing cached data to be evicted from a cache set.

32. (Currently Amended) The ~~apparatus~~ system of claim 31 wherein the dummy read instructions comprise instructions to read data from addresses in the main memory that do not store data transferred between the first and second processors.

33. (Currently Amended) The ~~apparatus~~ system of claim 31 wherein each cache set has n cache lines, n being an integer, and the first processor generates at least n dummy read instructions either

after generating one or more write instructions to write data intended for the second processor or

before generating one or more read instructions to read data that was written by the second processor.

34. (Currently Amended) The ~~apparatus~~ system of claim 31 wherein the main memory comprises a first portion that does not store data transferred between the first and second processors.

35. (Currently Amended) The ~~apparatus~~ system of claim 34 wherein the first portion of the main memory has a size that is at least the number of cache lines per cache set multiplied by the number of cache sets multiplied by the number of bytes per cache line.